

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-262025

(43)Date of publication of application : 13.10.1995

(51)Int.Cl.

G06F 9/46

(21)Application number : 06-049504

(71)Applicant : FUJITSU LTD

(22)Date of filing : 18.03.1994

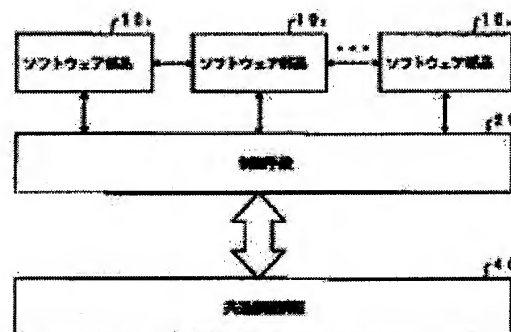
(72)Inventor : KAWAKAMI KUMIKO  
AKAGAWA YASUKO  
TAKAHASHI MASUMI

## (54) EXECUTION CONTROL SYSTEM

### (57)Abstract:

**PURPOSE:** To provide an execution control system for which it is not-necessary to change each application by independently managing application software or the like.

**CONSTITUTION:** This system is provided with common reference information 40 containing procedure information to be commonly referred to among software parts 101-10n when execution request information is issued from software parts 101-10n, and control means 20 for controlling execution among the software parts 101-10n by coupling the software parts while referring to the common reference information 40 corresponding to the execution request information issued from the software parts 101-10n.



(19) 日本国特許庁 ( J P )

(12) 公 開 特 許 公 報 ( A )

(11) 特許出願公開番号

特開平7-262025

(43) 公開日 平成7年(1995)10月13日

(51) Int.Cl.<sup>5</sup>

G 0 6 F 9/46

識別記号

庁内整理番号

F I

技術表示箇所

3 4 0 A 7737-5B

審査請求 未請求 請求項の数20 O L (全 25 頁)

(21) 出願番号 特願平6-49504

(22) 出願日 平成6年(1994)3月18日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72) 発明者 川上 久美子

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(72) 発明者 赤川 寧子

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(72) 発明者 高橋 ますみ

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74) 代理人 弁理士 伊東 忠彦

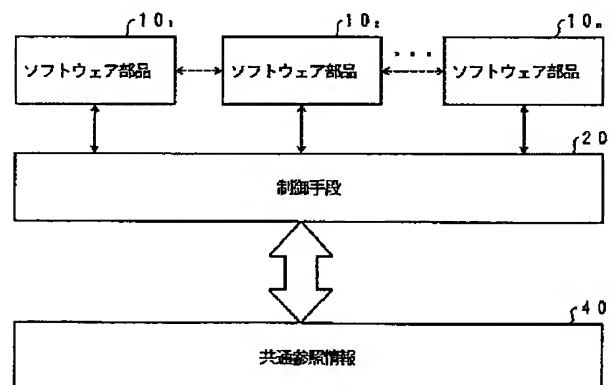
(54) 【発明の名称】 実行制御システム

(57) 【要約】

【目的】 本発明の目的は、アプリケーションソフト等を独立して管理し、アプリケーション個々の変更を行わなくてもよい実行制御システムを提供することである。

【構成】 本発明は、ソフトウェア部品10からの実行依頼情報が発行された場合にソフトウェア部品10間で共通に参照する手続情報を含む共通参照情報40と、ソフトウェア部品10から発行された実行依頼情報により共通参照情報40を参照し、ソフトウェア部品を結合し、ソフトウェア部品10間の実行を制御する制御手段20とを有する。

本発明の原理構成図



**【特許請求の範囲】**

【請求項1】 所定の規格により生成された複数のソフトウェア部品（10）間の実行制御、該ソフトウェア部品間のメッセージの送受信及び、該ソフトウェア部品（10）の実行を冗長系における主系または従系の運転モードのいずれかで実行する場合に、該ソフトウェア部品（10）から実行依頼情報が発行された場合に、該ソフトウェア部品（10）間で共通に参照する手続情報を含む共通参照情報（40）と、該ソフトウェア部品（10）から発行された実行依頼情報により該共通参照情報（40）を参照し、該ソフトウェア部品（10）を結合し、該ソフトウェア部品（10）間の実行を制御する制御手段（20）とを有することを特徴とする実行制御システム。

【請求項2】 前記制御手段（200）は、あるソフトウェア部品（10）より他のソフトウェア部品（10）への実行依頼情報（500）を受け取ると、該実行依頼情報（500）に基づいて前記共通参照情報（400）を参照し、参照された結果に基づいて、該実行依頼情報（500）に指定された宛先（520）に該実行依頼情報（500）を渡す請求項1記載の実行制御システム。

【請求項3】 前記共通参照情報（40）は、カスタマイズ時に指定する請求項1記載の実行制御システム。

【請求項4】 前記ソフトウェア部品間の実行依頼情報（500）は、カスタマイズ時に、他のソフトウェア部品の宛先情報（520）を付加して前記制御手段（200）に渡す請求項1記載の実行制御システム。

【請求項5】 前記制御手段（200）は、複数のソフトウェア部品を組み合わせる1プロセスに編集する、または、同一ソフトウェア部品を複数のプロセスに割り付けるように前記共通参照情報（400）をカスタマイズ時に指定する請求項1記載の実行制御システム。

【請求項6】 前記制御手段（200）は、依頼元の前記ソフトウェア部品より発行された前記実行依頼情報（500）を受け取り、該実行依頼情報（500）に基づいて、前記共通参照情報（400）を参照し、該依頼元のソフトウェア部品と依頼先のソフトウェア部品が同一プロセス内にある場合には、前記実行依頼情報（500）を関数呼出による引数渡しとし；該依頼元のソフトウェア部品と依頼先のソフトウェア部品が同一装置内に設けられ、かつ別プロセスの場合には、前記実行依頼情報（500）をメッセージキュー渡しとし；該依頼元のソフトウェア部品と依頼先のソフトウェア部品が別装置に設けられている場合には、前記実行依頼情報をソケット渡しとする請求項2記載の実行制御システム。

【請求項7】 前記制御手段（200）は、

依頼元のソフトウェア部品からの前記実行依頼情報（500）に基づいて実行した依頼先のソフトウェア部品からの処理結果の通知を該実行依頼情報（500）の指定に基づいて、

該実行依頼情報（500）に処理結果不要と指定されている場合には、該依頼元のソフトウェア部品に処理結果の通知を行わない；該依頼元のソフトウェア部品から同期要求による依頼の場合には、該依頼元のソフトウェア部品は該依頼先のソフトウェア部品からの処理結果が通知されるまで待ち状態となり、

該依頼元のソフトウェア部品と該依頼先のソフトウェア部品が同一プロセス内にある場合には、関数復帰時に引数渡しとする；該依頼元のソフトウェア部品と該依頼先のソフトウェア部品が同一装置内の別プロセスの場合には、メッセージキュー渡しとする；該依頼元のソフトウェア部品と該依頼先のソフトウェア部品が別装置にある場合には、ソケット渡しとする請求項2記載の実行制御システム。

【請求項8】 前記制御手段（200）は、依頼元のソフトウェア部品からの実行依頼情報（500）に基づいて実行した依頼先のソフトウェア部品からの処理結果の通知を該実行依頼情報（500）の指定に基づいて、

該依頼元のソフトウェア部品から非同期要求による依頼の場合には、該依頼元のソフトウェア部品は依頼先のソフトウェア部品からの処理結果の待ち合わせをせずに処理を続行し、該処理結果は別途通知することとし、該通知は、

該依頼元のソフトウェア部品と該依頼先のソフトウェア部品が同一装置内に設けられている場合には、メッセージキュー渡しとする；該依頼元のソフトウェア部品と該依頼先のソフトウェア部品が別装置に設けられている場合には、ソケット渡しとする請求項2記載の実行制御システム。

【請求項9】 前記共通参照情報（31）は、一連のメッセージの流れを業務とし、業務数分の情報が定義される請求項1記載の実行制御システム。

【請求項10】 前記制御手段（3）は、前記共通参照情報（31）を参照し、送信依頼側のソフトウェア部品（1）と受信側のソフトウェア部品（5）とのインタフェースの差異を吸収して、送信依頼側のソフトウェア部品と受信側のソフトウェア部品間のメッセージの通信を制御する請求項9記載の実行制御システム。

【請求項11】 前記制御手段（3）は、送信依頼側のソフトウェア部品（1）から受信したメッセージの形式を受信側のソフトウェア部品（5）の受信形式に変換して送信する請求項10記載の実行制御システム。

【請求項12】 前記制御手段（3）は、

送信依頼側のソフトウェア部品(1)から受信したメッセージを共通参照情報(31)を参照して編集し、受信側のソフトウェア部品(5)に転送する請求項10記載の実行制御システム。

【請求項13】 前記送信依頼側のソフトウェア部品(1)から送信するメッセージ(2)の宛先を前記制御手段(3)とする請求項11記載の実行制御システム。

【請求項14】 前記前記送信依頼側のソフトウェア部品(1)から送信するメッセージ(2)は、宛先、メッセージコード、メッセージ及びインタフェースからなる請求項11記載の実行制御システム。

【請求項15】 前記制御手段(3)は、送信依頼側のソフトウェア部品から複数のメッセージを受信して、受信側のソフトウェア部品に送信する請求項11記載の実行制御システム。

【請求項16】 前記制御手段(3)は、送信依頼側のソフトウェア部品から受信した1つのメッセージを受信して、該メッセージの前記インタフェースを編集して、受信側のソフトウェア部品の数分複写して送信する請求項11記載の実行制御システム。

【請求項17】 前記制御手段(3)は、送信依頼側のソフトウェア部品(1)からの一連の情報を一つの業務として扱い、業務開始メッセージ受付時から業務終了メッセージを受信するまで監視し、その間有効メッセージとして前記共通参照情報(31)に定義されている以外の情報を受け付けけない請求項11記載の実行制御システム。

【請求項18】 前記制御手段(3)は、送信依頼側のソフトウェア部品(1)からの一連のメッセージを一つの業務として扱い、前記共通参照情報(31)に定義されている監視時間(3004)まで該メッセージの処理を監視し、該監視時間中は、受信した情報を有効メッセージとして、前記共通参照情報(31)に定義されている以外の情報を受け付けけない請求項11記載の実行制御システム。

【請求項19】 前記共通参照情報(60)は、主系モード運転及び従系モード運転を行う冗長系において、ソフトウェア部品間の結合関係及び業務を組み合わせてステータスとして定義し、定義された該ステータスの有効/無効状態を管理し、前記制御手段(20)は、主系モード運転及び従系モード運転を行う冗長系において、前記共通参照情報(50)を参照して運転モードを切り替える請求項1記載の実行制御システム。

【請求項20】 前記制御手段(20)は、運転モードを切り替えた場合には、前記共通参照情報(50)のステータスの有効/無効情報を変更する請求項19記載の実行制御システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、実行制御システムに係り、特に、リアルタイム情報を扱うコンピュータシステムのソフトウェア開発において、所定の規格に従って作成されたソフトウェア群を組み合わせる処理要求に対応するためのソフトウェア部品間の結合方式、ソフトウェア部品間のメッセージ送受信、冗長系においてソフトウェア部品の実行時の運転モードの切替を行う運転モード制御方式等で利用するソフトウェア部品間における実行制御システムに関する。

【0002】近年、個々に開発されたソフトウェアを複数組み合わせるリアルタイムに処理を行う場合に、個々のソフトウェアの結合条件設定や、ソフトウェアの変更処理を行うというような処理を簡素化することが望まれている。

【0003】

【従来の技術】全てのソフトウェア部品は、プロセス/ライブラリに係わらず、処理依頼発生により処理を開始し、依頼機能を実現させて処理を終了する。ある処理依頼に対して、従来のソフトウェア部品間におけるメッセージ等の受渡しの方法等について以下に示す。

【0004】図20は、従来の第1の例を示す。同図に示すシステムにおいて装置11は、2つのプロセス1、2を有し、プロセス1は、ソフトウェア部品a、ソフトウェア部品bを含み、プロセス2は、ソフトウェア部品cを含む。装置13は、1つのプロセス3を有し、プロセス3は、ソフトウェア部品dを含む。ここで、ソフトウェア部品とは、任意の処理を行うアプリケーションプログラムを指す。

【0005】従来のシステムは、複数のソフトウェア部品動詞を結合する場合には、結合方法、結合相手等の結合情報がソフトウェア(プログラム)内に生成されている。

【0006】図20に示すように、ソフトウェア部品aは、ソフトウェア部品bに外部から受け取ったメッセージを転送するロジックが組み込まれ、メッセージを受け取るソフトウェア部品bは、ソフトウェア部品aからメッセージを受け取り、そのメッセージをプロセス2のソフトウェア部品cに転送するロジックが組み込まれている。または、ソフトウェア部品bからソフトウェア部品cにメッセージを転送する以外に、プロセス2を起動するロジックまたは、テーブル/ファイル渡しの場合もある。さらに、プロセス2のソフトウェア部品cから装置13のプロセス1のソフトウェア部品dに対して、情報の書込みは書き込む情報をソケット渡しにより行われる。ソフトウェア部品dは、書き込み情報をソフトウェア部品cより読み込むロジックが含まれている。また、装置11から装置13へ書き込み情報のソケット渡しを行う他、個々のインタフェースを規定する場合もある。

【0007】図20に示すように、各ソフトウェア部品a、b、c、dは、“メッセージ受信”、“メッセージ

送信”、“メッセージ書込み”、及び“メッセージ読み込み”等の他のソフトウェア部品に実行を依頼するための結合情報（実行依頼情報）を内部に有する。

【0008】図21は、従来の第2の例を示す図である。第2の例は、メッセージをメッセージ管理部14に集め、宛先対象表15を用いて仮定の宛先から実宛先に変換して送信する方法（特開平3-191429）である。

【0009】第2の例において、メッセージ管理部14は、宛先対象表15を参照して、宛先のみを変換する。メッセージのインタフェースは、ソフトウェア部品（PKG）のプログラム内部で静的に記述する。例えば、メッセージの待ち合わせ等の同期制御は、ソフトウェア部品のプログラム内部の処理で行う。

【0010】同図（A）において、ソフトウェア部品PKG-Bからソフトウェア部品PKG-Aにメッセージa及びメッセージbの2つのメッセージが送信されている。メッセージは、メッセージ管理部14を宛先とする宛先情報と、仮想宛先、及びインタフェース及びメッセージ内容等で構成される。

【0011】ソフトウェア部品PKG-Bより送出されるメッセージaは、仮想宛先を“001”とする。従って、メッセージ管理部14は、メッセージaの仮想宛先（001）に基づいて、宛先対象表15を検索する。検索によりメッセージの宛先は、PKG-Aであることを検出する。また、メッセージ管理部14は、メッセージbの仮想宛先（001）に基づいてメッセージaと同様に、宛先対象表15を検索し、メッセージの宛先がPKG-Aであることを検出し、メッセージbの内容をソフトウェア部品PKG-Aに転送する。

【0012】また、ソフトウェア部品PKG-Bから転送されるメッセージのうち、メッセージbの宛先をソフトウェア部品PKG-Cに変更した場合には、図21（B）に示すように、PKG-Bのプログラムを修正して宛先対象表15の仮想宛先を“001”から“003”へ変更する。この場合は、宛先となるソフトウェア部品のメッセージのインタフェースを変えるためにプログラムの修正を行う。

【0013】図22は、従来の第2の例においてプログラム修正により宛先を変更する例を示す。ソフトウェア部品PKG-BからPKG-Aに送信しているメッセージの宛先をソフトウェア部品PKG-Cに変更する場合には、ソフトウェア部品PKG-Cが受け取るべき情報を送信するようにソフトウェア部品PKG-Bの送信内容をプログラム変更により修正する。

【0014】図22（A）は、ソフトウェア部品PKG-Aがソフトウェア部品PKG-Bからのメッセージを受け取れるように、インタフェースを変更し、仮想宛先を“001”としてメッセージを送出する場合を示す。同図（B）は、ソフトウェア部品PKG-Bから送出す

る宛先をソフトウェア部品PKG-Cに変更する場合には、ソフトウェアPKG-Cが当該メッセージを受け取ることができるように、宛先に対応するインタフェースに書き換える。

【0015】次に、従来のシステムの運転モード管理について説明する。運転モード管理は、従来、アプリケーションにより実行され、運転モードの切替が発生した場合に、切替られたモードで処理を行う。このとき、各業務の中で通常運転モードならば、取り込んだデータを集計して出力する。一方試験モード運転の場合は、取り込んだデータは廃棄し、テストデータで集計を行うといったような自系の運転モードの管理を行い自系のモードを常に意識し、運転モードに合わせて処理を随時選択する。

【0016】図23は、従来のシステムの運転モード管理システムを説明するための図である。

【0017】運転モードの決定を行う場合に、アプリケーションソフトは、モード切替要求が入力されると（ステップ1）、自系の運転モードを参照し、運転モードが通常運転モードか試験運転モードであるかを判定する。最初に、アプリケーションソフトは、試験運転モードであるかを自系モードを参照して判定する（ステップ2）。この結果、試験運転モードであると判定された場合には、試験運転モードで処理を実行し、実行中に取り込んだデータを破棄し、テストデータで集計を行う（ステップ3）。一方、通常運転モードであるかを自系のモードを参照して判定し（ステップ4）、通常運転モードであれば、実行中に取り込んだデータを集計して出力する（ステップ5）。また、通常運転モードでも試験運転モードでもない場合には、それら以外のモードの処理を行うか、またはエラーとする（ステップ6）。

【0018】

【発明が解決しようとする課題】しかしながら、上記従来の方法には、以下に示すような問題がある。

【0019】まず、第1の従来の例は、結合方法や、結合相手等の結合情報がプログラム内に設定されているため、装置／プロセス構成、結合相手を変更する場合には、プログラム改造が必要となる。また、第1の従来の例は、全てのケースで可能な結合方法（ソケット）とした場合には、OSオーバヘッド増による大幅な性能低下がおきる。また、結合情報のをプログラム内に設定するために難解なソフト構造となり、第三者が変更等を行うのが困難である。

【0020】さらに、第2の従来の例は、ある宛先に送信していたメッセージの宛先を変更する場合にはプログラムを変更するため、ソフトウェア部品の独立性を低下させるという問題がある。

【0021】また、第2の従来の例は、メッセージの待ち合わせを行っているソフトウェア部品で、待ち合わせするメッセージの数や、待ち合わせていないメッセージが

到来したときの処理、待ち合わせる時間等を変更する時にプログラムを修正しなければならないという問題がある。

【0022】また、第2の従来例は、メッセージの送信元や、送信される内容に変更があった場合、インタフェースの分配も変更しなければならないが、この変更はプログラムを修正する必要がある、ソフトウェア部品の独立性を低下させるという問題がある。

【0023】また、第2の従来例は、複数のメッセージの管理をソフトウェア部品内部で行っていたため、複数のソフトウェア部品に対して起動のメッセージが送信されてしまっても監視できない。また、一連のメッセージの流れの中の一箇所でも変更があった場合、関係するすべてのソフトウェア部品において、プログラム修正を行わなければならない、ソフトウェア部品の独立性を低下させるという問題がある。

【0024】また、メッセージ管理部において、宛先対照表の宛先は変更できるがソフトウェア部品によってインタフェースが異なるため、宛先に応じたインタフェースに書き換える必要がある。

【0025】さらに、従来の二重化システムの運転モード管理システムは、運転モードの切替が発生した場合に、アプリケーションソフトで管理を行うため、システム移行や、運転モードの追加等を行う場合には、アプリケーションソフトを改造する必要がある。

【0026】本発明の概括的な目的は、アプリケーションソフト等を独立して管理し、アプリケーション個々の変更を行わなくてもよい共通情報参照システムを提供することである。

【0027】本発明の詳細な目的は、ソフトウェア部品が独立に開発でき、複数のソフトウェア部品を結合させて使用する場合に、個々のソフトウェア部品の変更を行わずに、ソフトウェア部品間のメッセージ送受信管理が可能なソフトウェア部品結合方式を提供することである。

【0028】また、本発明の更なる詳細な目的は、運転モードの追加やシステム移行がアプリケーションソフトの改造を行わなくても可能な運転モード管理システムを提供することである。

【0029】また、本発明の更なる詳細な目的は、送信ソフトウェア部品が宛先ソフトウェア部品を意識せずにメッセージを送信することが可能なメッセージ送受信システムを提供することである。

【0030】

【課題を解決するための手段】本発明の実行制御システムにおけるソフトウェア部品結合方式は、所定の規格により生成された複数のソフトウェア部品10間の実行制御、ソフトウェア部品間のメッセージの送受信及び、ソフトウェア部品10の実行を冗長系における主系または従系の運転モードのいずれかで実行する場合に、ソフト

ウェア部品10からの実行依頼情報が発行された場合にソフトウェア部品10間で共通に参照する手続情報を含む共通参照情報40と、ソフトウェア部品10から発行された実行依頼情報により共通参照情報40を参照し、ソフトウェア部品を結合し、ソフトウェア部品10間の実行を制御する制御手段20とを有する。

【0031】また、上記の制御手段200は、あるソフトウェア部品10より他のソフトウェア部品への実行依頼情報500を受け取ると、実行依頼情報500で共通参照情報400を参照し、参照された結果に基づいて、実行依頼情報500に指定された宛先520に実行依頼情報500を渡す。

【0032】また、上記の共通参照情報40は、カスタマイズ時に指定する。

【0033】また、ソフトウェア部品間の実行依頼情報500は、カスタマイズ時に、他のソフトウェア部品の宛先情報520を付加して制御手段200に渡す。

【0034】また、上記の制御手段200は、複数のソフトウェア部品を組み合わせる1プロセスに編集する、または、同一ソフトウェア部品を複数のプロセスに割り付けるように共通参照情報400をカスタマイズ時に指定する。

【0035】また、上記の制御手段200は、依頼元のソフトウェア部品より発行された実行依頼情報500を受け取り、実行依頼情報500に基づいて、共通参照情報400を参照し、依頼元のソフトウェア部品と依頼先のソフトウェア部品が同一プロセス内にある場合には、実行依頼情報500を関数呼出による引数渡しとし；依頼元のソフトウェア部品と依頼先のソフトウェア部品が同一装置内であり、かつ別プロセスの場合には、実行依頼情報500をメッセージキュー渡しとし；依頼元のソフトウェア部品と依頼先のソフトウェア部品が別装置である場合には、実行依頼情報をソケット渡しとする。

【0036】また、上記の制御手段200は、依頼元のソフトウェア部品からの実行依頼情報500に基づいて実行した依頼先のソフトウェア部品からの処理結果の通知を実行依頼情報500の指定に基づいて、実行依頼情報500に処理結果不要と指定されている場合には、依頼元のソフトウェア部品に処理結果の通知を行わない；依頼元のソフトウェア部品から同期要求による依頼の場合には、依頼元のソフトウェア部品は依頼先のソフトウェア部品からの処理結果が通知されるまで待ち状態となり、依頼元のソフトウェア部品と依頼先のソフトウェア部品が同一プロセス内にある場合には、関数復帰時に引数渡しとする；依頼元のソフトウェア部品と依頼先のソフトウェア部品が同一装置内の別プロセスの場合には、メッセージキュー渡しとする；依頼元のソフトウェア部品と依頼先のソフトウェア部品が別装置にある場合には、ソケット渡しとする。

【0037】また、上記の制御手段200は、依頼元の

ソフトウェア部品からの実行依頼情報500に基づいて実行した依頼先のソフトウェア部品からの処理結果の通知を実行依頼情報500の指定に基づいて、依頼元のソフトウェア部品から非同期要求による依頼の場合には、依頼元のソフトウェア部品は依頼先のソフトウェア部品からの処理結果の待ち合わせをせずに処理を続行し、処理結果は別途通知することとし、通知は、依頼元のソフトウェア部品と依頼先のソフトウェア部品が同一装置内にある場合には、メッセージキュー渡しとする；依頼元のソフトウェア部品と依頼先のソフトウェア部品が別装置にある場合には、ソケット渡しとする。

【0038】また、本発明の実行制御システムにおけるメッセージ送受信方式の共通参照情報31は、一連のメッセージの流れを業務とし、業務数分の情報が定義される。

【0039】また、上記の制御手段3は、共通参照情報31を参照し、送信依頼側のソフトウェア部品1と受信側のソフトウェア部品5とのインタフェースの差異を吸収して、送信依頼側のソフトウェア部品と受信側のソフトウェア部品間のメッセージの通信を制御する。

【0040】また、上記の制御手段3は、送信依頼側のソフトウェア部品1から受信したメッセージの形式を受信側のソフトウェア部品5の受信形式に変換して送信する。

【0041】また、上記の制御手段3は、送信依頼側のソフトウェア部品1から受信したメッセージを共通参照情報31を参照して編集し、受信側のソフトウェア部品5に転送する。

【0042】また、上記の送信依頼側のソフトウェア部品1から送信するメッセージ2の宛先を制御手段3とする。

【0043】また、上記の送信依頼側のソフトウェア部品1から送信するメッセージ2は、宛先、メッセージコード、メッセージ及びインタフェースからなる。

【0044】また、上記の制御手段3は、送信依頼側のソフトウェア部品から複数のメッセージを受信して、受信側のソフトウェア部品に送信する。

【0045】また、上記の制御手段3は、送信依頼側のソフトウェア部品から受信した1つのメッセージを受信して、メッセージのインタフェースを編集して、受信側のソフトウェア部品の数分複写して送信する。

【0046】また、上記の制御手段3は、送信依頼側のソフトウェア部品1からの一連の情報を一つの業務として扱い、業務開始メッセージ受付時から業務終了メッセージを受信するまで監視し、その間有効メッセージとして共通参照情報31に定義されている以外の情報を受け付けない。

【0047】また、上記の制御手段3は、送信依頼側のソフトウェア部品1からの一連のメッセージを一つの業務として扱い、共通参照情報31に定義されている監視

時間3004までメッセージの処理を監視し、監視時間中は、受信した情報を有効メッセージとして、共通参照情報31に定義されている以外の情報を受け付けない。

【0048】さらに、本発明の実行制御システムにおける運転モード制御方式において、共通参照情報60は、主系モード運転及び従系モード運転を行う冗長系において、ソフトウェア部品間の結合関係及び業務を組み合わせでステータスとして定義し、定義されたステータスの有効／無効状態を管理し、制御手段20は、主系モード運転及び従系モード運転を行う冗長系において、共通参照情報50を参照して運転モードを切り替える。

【0049】また、上記の制御手段20は、運転モードを切り替えた場合には、共通参照情報50のステータスの有効／無効情報を変更する。

【0050】

【作用】本発明は、リアルタイム情報を扱うコンピュータシステムのソフトウェア開発において、共通に参照できる情報を持つことにより、規格に従って作成されたソフトウェア部品群を自由に組み合わせて実行することが可能となる。

【0051】また、本発明は、使用するソフトウェア部品、ソフトウェア部品を有するプロセスをカスタマイズ時に共通参照情報に定義することにより、指定ソフトウェア部品の搭載や、各ソフトウェア部品の動作環境の設定、各プロセスの動作制御を行うことが可能なり、ソフトウェア部品のプログラムを改造せずに、業務に合致したコンピュータシステムを構築することが可能となる。

【0052】さらに、本発明は、システム内の各プロセスについて構成ソフトウェア部品をカスタマイズ時に定義することにより、プログラム改造を必要とせずに、複数のソフトウェア部品を組み合わせで1つのプロセスに編集したり、同一ソフトウェア部品を複数のプロセスに割り振ったりすることができ、新規にプログラムを構築する必要がない。

【0053】また、ソフトウェア部品間の実行依頼情報の受渡しをカスタマイズ時に宛先情報を実行依頼情報に付加することにより、目的とするシステムの処理シーケンスに合わせてプログラム改造を行わずに、各ソフトウェア部品と任意に指定されたソフトウェア部品とを接続することができるため、個々のソフトウェア部品内でインタフェースを接続する相手のソフトウェア部品用に修正する等の作業が不要となる。

【0054】また、本発明において、ソフトウェア部品間は、全て同一手続で結合（実行依頼の受渡し）し、制御手段において、カスタマイズ時に定義された結合相手のソフトウェア部品の環境に合わせて最適な方法で実行依頼情報を渡すことができる。

【0055】また、本発明は、実行依頼先のソフトウェア部品で処理された結果を依頼元のソフトウェア部品に通知する場合に、依頼時の実行依頼情報に指定に従っ



て、同一手続とすることにより、制御手段が自動的に最適な方法で処理結果通知を発行し、転送することが可能となる。

【0056】さらに、本発明のメッセージ送受信システムにおいては、メッセージで通知したいインタフェース情報の形式をデータ化して、制御手段で編集し、相手のソフトウェア部品のインタフェースに変換することができるため、送信側のソフトウェア部品は、相手のインタフェース情報を認識していなくとも、メッセージの送信が可能となる。

【0057】また、制御手段において、複数の送信側のソフトウェア部品からメッセージを受け取り、1つのソフトウェア部品に送信する場合に、制御手段が複数のメッセージを宛先情報に基づいて1つのメッセージに編集して転送することにより、メッセージ送信のトラヒック数が削減される。

【0058】また、1つの送信側のソフトウェア部品からのメッセージを複数のソフトウェア部品に転送する場合に、制御手段において、分割編集して転送することにより、同報システム等において1つのソフトウェア部品内部で送信相手のソフトウェア部品数の処理を繰り返す必要がなく、送信相手先の数の増減があっても共通参照情報を変更するのみでよい。

【0059】また、一連のメッセージを一つの業務という単位で扱い、業務開始のメッセージ受信時から業務終了のメッセージを受信するまで、または、共通参照情報に定義されている時間まで監視し、その間有効メッセージとして共通参照情報に定義されている以外のメッセージを受け付けられないという排他制御が可能であり、該当業務処理の二重起動を防止することが可能となる。

【0060】さらに、本発明の運転モード管理システムは、二重化システムにおける主系モード運転/従系モード運転やメンテナンス時の試験モード運転のように、運転モードにより行う処理シーケンス（データフロー）が異なる場合に運転モードに合わせて自動的に処理シーケンスを切り替え、ソフトウェア部品での運転モード認識が不要となる。

【0061】

【実施例】以下、図面とともに本発明の実施例を説明する。

【0062】[第1の実施例] 図2は、本発明の第1の実施例のソフトウェア部品の結合システムの概要を示す。同図に示すシステムは、プロセスX、プロセスYを有する装置α、プロセスZを有する装置βと結合制御部200より構成される。結合制御部200は、イベント結合ルーチン210及び通知情報域220を含む。プロセスAは、イベント情報101を含む部品A、イベント102を含む部品Bを有する。プロセスYは、イベント情報103を含む部品Cを有する。装置βのプロセスZは、部品Dを有する。

【0063】ソフトウェア部品A、B、D、Dは、図3に示すように、全て、イベント情報500及び応答イベント格納領域600からなる引数を有し、入力イベント情報に従って処理を実行する。他のソフトウェア部品へのイベント情報がある場合には、結合制御部200にイベント結合ルーチン210をサブルーチンコールする。この時の引数は、イベント情報500及び応答イベント格納領域600である。また、イベント情報を受け取り実行したソフトウェア部品が発信元のソフトウェア部品に渡す処理結果がある場合には、結合制御部200の応答イベント格納領域の通知情報域220に処理結果を格納する。また、イベント結合ルーチン210の呼出元に復帰する場合に、ソフトウェア部品からのイベント情報500の復帰コードにより判定して復帰する。復帰コードは例えば、“0”の場合には、処理結果を無しとし、“1”の場合には、処理結果情報長があるとする。

【0064】図4は、本発明の一実施例のイベント情報形式を示す。イベント情報500はイベント結合ルーチン210をコールする際に、引数として用いられる。他のソフトウェア部品へ情報渡しも、イベント情報を引数としてイベント結合ルーチン210を呼ぶことにより実現される。イベント情報500の形式は、送信イベント及び応答イベントとも共通であり、イベント種別510、装置ID521、プロセスID522及びソフトウェア部品ID523をイベントを送る宛先520、装置ID531、プロセスID532、ソフトウェア部品ID533及び、イベント番号534を発信元情報530、通知情報長540、及び通知情報550より構成される。このうち、宛先情報520の装置ID521、プロセスID522、ソフトウェア部品ID523、発信元情報530の装置ID531、プロセスID532は発信元のソフトウェア部品で設定する必要はなく、イベント結合ルーチン210が結合情報400を参照して設定される。また、イベント種別510は処理結果不要、同期要求、非同期要求等が設定される。

【0065】結合制御部200は、リアルタイム情報及びバッチ情報300が入力されると、その情報300を処理するプロセスのイベントに出力する。また、各プロセス間の結合情報400を参照し、プロセス間の結合情報のあるプロセスから他のプロセスに渡す制御を行う。なお、結合情報400は、ソフトウェア部品内に生成せず、外部情報としてカスタマイズ時に指定するものとする。

【0066】図5は、本発明の第1の実施例の結合情報の例を示す。結合情報400は、ソフトウェア部品の定義、プロセス定義、結合相手情報から構成される。ソフトウェア部品の定義において、その使い方は、各ソフトウェア部品毎のカスタマイズ項目があるときに指定される。プロセス定義は、プロセスごとのソフトウェア部品とその属性が指定される。このうち、属性は常駐/非常



駐動作レベル等を指定するものである。結合相手情報は、依頼種別と宛先から構成される。

【0067】結合制御部200は、リアルタイム情報及びバッチ情報300が入力されると、例えば、プロセス内の結合手続関数がイベント110に出力される。結合制御部200は、プロセスの部品aから同じプロセスの部品bに渡す依頼情報イベント101が入力されると、結合手続関数がイベント111を介して同一プロセスの部品bに関数で渡す(関数渡しr)。また、結合制御部200は、プロセスの部品bからプロセスYに対してはメッセージの形で、イベント112、113を介して渡す(メッセージ渡しs)。さらに、装置αのプロセスYから装置βのプロセスZに渡す依頼情報イベント103が入力されると、ソケット通信の形でプロセスZにイベント115を介して渡す。

【0068】図6は、本発明の第1の実施例の結合制御部の動作の概要を示すフローチャートであり、図7は、図6のフローチャートの動作に対応する動作を説明するための図である。図7の番号は図6のステップ番号に対応しており、説明上、網掛け部分は、結合制御部200の動作を示している。

【0069】ステップ100) 他の装置または他のプロセスからイベントがプロセスXのソフトウェア部品Cに発行される。

【0070】ステップ101) 結合制御部200は、プロセスXのメッセージキュー100から発行されたイベント100を取り出す。

【0071】ステップ102) 次に、結合制御部200は、取り出されたイベントを引数としてイベント結合ルーチン210を呼ぶ。

【0072】ステップ103) イベント結合ルーチンはイベントに指定されているソフトウェア部品の宛先が自プロセス内であるか他のプロセスであるのかを結合情報400を参照して判断する。自プロセス内に宛先部品がある場合には、ステップ104に移行し、自プロセス内に宛先部品がない場合には、ステップ107に移行する。

【0073】ステップ104) 宛先のソフトウェア部品が自プロセス内にある場合には、宛先で指定されている自プロセス内のソフトウェア部品を呼び出す。図7の例では、宛先がソフトウェア部品Aとなっている場合には、プロセスX内にある宛先部品(ソフトウェア部品A)が自プロセス内にあるので、イベントを引数としてソフトウェア部品Aを呼び出し関数渡しとする。

【0074】ステップ105) ソフトウェア部品Aが実行され、他にイベント情報があるかを判定し、ない場合にはステップ101に移行する。他にイベント情報がある場合には、ステップ106に移行する。

【0075】ステップ106) 結合制御部200は、イベント情報に結合情報400の宛先情報を設定してス

テップ102に移行する。

【0076】ステップ107) 宛先は、他装置のソフトウェア部品かを判定する。宛先が他装置のソフトウェア部品である場合には、ステップ110に移行する。宛先が同一装置である場合には、ステップ108に移行する。

【0077】ステップ108) 宛先プロセスのメッセージキューにイベント情報をキューイングする。図7の例では、プロセスXから他プロセスのメッセージキュー140にメッセージ渡しする。

【0078】ステップ110) 宛先が他装置にあるソフトウェア部品である場合には、宛先装置向けの回線にイベント情報をソケット送信する。また、他の装置ηは、回線を介して他装置βから送信されたイベント情報を受信し、イベント情報の宛先により該当プロセスのメッセージキューへイベント情報を送信する。

【0079】次に、カスタマイズ時に結合制御部200が定義する結合情報について説明する。

【0080】まず、カスタマイズ時に設定する項目として、該当装置に接続するための他の装置の一覧、該当装置のプロセス一覧、及びソフトウェア部品のイベント宛先情報一覧を定義する。

【0081】(1) まず、該当装置に接続するための情報である他装置の一覧は、表1に示すように装置IDと接続回線IDから構成される。

【0082】

【表1】

他装置の一覧

| 装置ID | 接続回線ID |
|------|--------|
| 装置α  | 自装置    |
| 装置β  | 回線5    |
| 装置η  | 回線2    |

【0083】表1に示す他装置の一覧は、イベント結合ルーチン210がこの情報を参照してイベント情報の宛先装置が自装置かまたは他装置かを判断する。また、この情報を参照してイベント情報の宛先装置向け回線を決定し、イベント情報を出力する。

【0084】(2) 該当装置のプロセス一覧は、表2に示すように、プロセスIDと構成ソフトウェア部品IDより構成される。

【0085】

【表2】

該当装置プロセス一覧

| プロセスID | 構成ソフト 部品ID |
|--------|------------|
| プロセスX  | 部品A 部品B    |
| プロセスY  | 部品C        |

【0086】表2に示す該当装置のプロセス一覧は、結合制御部200が各プロセス生成時に、この一覧を参照し、どのソフトウェア部品を組み合わせたプロセスを生成するかを判定する。

【0087】(3)ソフトウェア部品イベント宛先情報

ソフトウェア部品イベント宛先情報一覧

|   | イベント種別      | 宛先              |
|---|-------------|-----------------|
| a | 部品Aからのイベント1 | 装置α, プロセスX, 部品B |
| b | 部品Aからのイベント2 | 装置α, プロセスY, 部品C |
| c | 部品Bからのイベント1 | 装置β, プロセスZ, 部品E |
| d | 部品Aからのイベント3 | 装置β, プロセスZ, 部品D |

【0089】イベント結合ルーチン210が、ソフトウェア部品よりイベント送信依頼を受けた時、イベント情報の発信元ソフトウェア部品IDと、イベント番号によりこの一覧を参照し、イベント情報内の宛先をセットする。

【0090】以下に、上記の結合情報を用いてソフトウェア部品間、プロセス間及び装置間で行うイベント情報の受渡しの方法を説明する。

【0091】ソフトウェア部品間でイベント情報を渡す場合には、前提として、自装置、自プロセス内で授受される場合について説明する。このとき、引数をイベント情報として呼び出された結合制御部200のイベント結合ルーチン210は、ソフトウェア部品イベント宛先情報一覧を参照する。イベント情報のイベントNoが図7に示す『ソフトウェア部品Aのイベント1』であれば、そのイベント情報の宛先は、“装置αのプロセスXに含まれる部品B”である。従って、図7の例では、ソフトウェア部品Aとソフトウェア部品Bは、プロセスXに含まれており、ソフトウェア部品Aからソフトウェア部品Bをコールする。

【0092】次に、あるソフトウェア部品から他のプロセスのソフトウェア部品をコールするイベント情報が発行された場合について説明する。ソフトウェア部品AからプロセスYをコールする場合には、ソフトウェア部品Aからのイベント2が発行されると、“装置α、プロセスY”を参照すると、コールするソフトウェア部品はソフトウェア部品Cということになる。このとき、プロセスYのメッセージキューへイベント情報を送信する。

【0093】次に、あるソフトウェア部品から他の装置のソフトウェア部品をコールするイベント情報が発行された場合について説明する。ソフトウェア部品Aから装置βをコールするイベント3が発行されると、イベント結合ルーチン210は、上記表3のソフトウェア部品イベント宛先情報一覧を参照する。その結果、装置βのプロセスZ、ソフトウェア部品Dをコールする。これにより、イベント結合ルーチン210は、装置β向け回線のソケットにイベント情報を送信する。このとき、表1

一覧は、表3に示すように、イベント種別と宛先により構成される。

【0088】

【表3】

を参照して目的装置に接続する接続回線IDを得る。装置βの場合には、回線5を使用して、イベント情報を送信する。装置βのプロセスZは、メッセージキューよりイベントを取り出す。但し、メッセージがキューイングされていない場合には、キューイングされるまで待機する。次に、取り出したイベント情報の宛先により当該装置向けであるか他装置向けであるかを確認する。宛先が他装置になっている場合には、上記と同様に、他装置と接続される際の回線IDを参照してイベント情報を送信する。また、宛先が自装置になっている場合には、該当するソフトウェア部品をサブルーチンコールして実行する。処理が終了すれば、呼出元に自動的に復帰する。

【0094】ソフトウェア部品は、処理依頼発生時にイベント結合ルーチン210からの呼び出しにより動作し、実行が終了した場合に、処理結果を作成して呼出元に復帰する。このとき、イベント情報形式のイベント種別を参照して、処理結果が依頼元のソフトウェア部品で必要であるか否か、また、必要な場合には同期要求をしているか否か、また、通知方法を決定する。決定方法には以下のような方法がある。

【0095】・イベント情報の発行元のソフトウェア部品が処理結果を不要とする場合には、結合制御部200は、処理結果の通知を行わない。

【0096】・同期要求で依頼時は、イベント情報発行元のソフトウェア部品は、依頼結果が通知されるまで待ち状態となる。このときの通知方法として、発行元と依頼先のソフトウェア部品が同一プロセス内の場合、関数復帰時の引数渡しとする。また、発行元と依頼先のソフトウェア部品が同一装置内別プロセスの場合、メッセージキュー渡しとする。また、発行元と依頼先のソフトウェア部品が別装置の場合には、ソケット渡しとする。

【0097】・非同期要求で依頼時は、発行元は依頼結果を待ち合わせせずに、処理を続行し、依頼結果は別途通知される。このときの通知方法として、発行元と依頼先のソフトウェア部品が同一装置内の場合には、メッセージキュー渡しとする。また、発行元と依頼先のソフトウェア部品が別装置の場合、ソケット渡しとする。

【0098】[第2の実施例] 本発明の第2の実施例としてソフトウェア部品間のメッセージの送受信について説明する。

【0099】図8は、本発明の第2の実施例のメッセージ送受信システムの構成を示す。送信元ソフトウェア部品1が送信するメッセージ2は、宛先（業務処理シーケンス）、メッセージコード、インタフェースからなる。また送信元ソフトウェア部品1はテーブルa、bを有し、これらのテーブルは、送信のインタフェースを有する。送信元ソフトウェア部品1が送信するメッセージ2の宛先は常に業務処理シーケンス3であり、メッセージコードは、例えば、“001”としてその情報が業務処理シーケンスの定義体31に定義されている。

【0100】図9は、本発明の第2の実施例のメッセージ送受信のシーケンスチャートを示す。まず、業務処理シーケンス3がメッセージ2を送信元ソフトウェア部品1から受信する。メッセージの内容は、業務処理シーケンス3宛であり、そのメッセージコードは“001”である。インタフェースにはテーブルa、テーブルbが用いられているとする（ステップ30）。業務処理シーケンス3は、メッセージコード“001”を用いて定義体31を参照し（定義項目詳細30は、定義対内のメッセージコード001に関する定義情報の詳細内容例）、メッセージの形式変換を行う（ステップ31）インタフェースとして送られてきた2つのテーブルから業務処理シーケンス定義体31に定義してある部分を抜き出し、送信インタフェースとして編集しなおす（ステップ32）。これを受信側ソフトウェア部品5に送信するメッセージ4の形式に変換する（ステップ33）。メッセージ4の形式は、宛先、メッセージコード、インタフェースより構成され、宛先は、受信側ソフトウェア部品5（PKG-A）であり、メッセージコードは“101”として送信する（ステップ34）。

【0101】図10は、本発明の第2の実施例の定義体の構成を示す。定義体31は、一連のメッセージの流れとして業務をいくつか定義するかを設定する業務数3001と業務数3001文の業務情報を定義する業務情報3002より構成される。業務情報3002には、1から順に割り付ける業務番号3003、業務番号3003毎の業務監視を行う時間である業務監視時間3004、業務の開始の契機となるメッセージ情報を定義する業務開始メッセージ情報3005、業務の終了を示唆するメッセージ情報を定義する業務終了メッセージ情報3040、種々のメッセージの情報を定義するメッセージ情報3041から構成される。

【0102】このうち、業務開始メッセージ情報3005は、待ち合わせ時間3006、受信メッセージ数3009及び受信メッセージ詳細情報3010からなる受信メッセージ情報3008、送信メッセージ数3014、送信メッセージ詳細情報3015からなる送信メッセー

ジ情報3013より構成される。

【0103】受信メッセージ情報3008の待ち合わせ時間3006は、受信するメッセージが複数である場合には、待ち合わせ時間3006を適当な分数定義する。排他種別3007は、待ち合わせ中に待ち合わせていないメッセージを受信した時にこのメッセージを受け取るか、廃棄するかを定義する。受信するメッセージが単数の場合には、待ち合わせ時間3006と排他種別3007は定義しない。受信メッセージ情報3008は受信する全メッセージの情報を定義する。受信メッセージ数3009は、受信するメッセージの数定義する。受信メッセージ詳細情報3010は受信メッセージ数3009に定義した数だけ定義する。受信メッセージコード3011は、ここで受信すべきメッセージコードを定義する。メッセージ判定情報3012は、受信メッセージコード3011のメッセージの内容が正しいかどうかを判定する情報を定義する。

【0104】送信メッセージ情報3013には、送信する全メッセージの情報を定義する。送信メッセージ数3014は、送信するメッセージの数を定義する。送信メッセージ詳細情報3015は、送信メッセージ数3014に定義された数を定義する。宛先情報3016は、送信するメッセージのあてさきに関する情報を定義する。正常時情報3017は、受信したメッセージが正しかった時に送信するメッセージの情報を定義する。異常時情報3018は、受信したメッセージが異常であった時に送信するメッセージの情報を定義する。メッセージコード3019は、送信するメッセージのメッセージコードである。インタフェース編集情報3020は、受信したメッセージのインタフェースを編集する場合、編集方法等の情報を定義する。業務終了メッセージ情報では、業務監視の終了の契機となるメッセージ情報を定義する。メッセージ情報では、一般のメッセージ送受信に関する情報を定義する。定義項目や内容は業務開始メッセージ情報と同じである。

【0105】《業務開始メッセージ受信動作》次に、ソフトウェア部品が業務開始メッセージを受信した場合の動作を説明する。図11は、本発明の第2の実施例の業務開始メッセージを受信した場合の業務処理シーケンスの動作を示すフローチャートである。

【0106】業務処理シーケンス3は、業務監視外にメッセージを受信したとき（ステップ201）、そのメッセージコードが業務開始メッセージ情報3005の受信メッセージコード3011と同じであり（ステップ202）、待ち合わせ時間3006が0であり（＝待ち合わせを行わない）（ステップ203）、メッセージ判定情報3012によりそのメッセージが正常である（ステップ204）ことが判明したら、業務処理シーケンス3は、送信メッセージ数3014で定義したメッセージ数分のメッセージを送信メッセージ詳細情報3015に基

づいて送信する（ステップ206）。送信メッセージ内容は受信したメッセージは正常だったので、宛先情報3016の宛先に正常時情報のメッセージコード3019で、受信したインタフェースをインタフェース編集情報3021を基に編集する（ステップ205）。インタフェース編集が終了したらメッセージを送信する（ステップ206）。

【0107】また、業務番号3003に定義された番号の業務の業務監視時間3004までの監視を開始する（ステップ207）。

【0108】《業務終了メッセージ受信時動作》次に、ソフトウェア部品が業務終了メッセージを受信した場合の動作を説明する。図12は、本発明の第2の実施例の業務終了メッセージを受信した場合の業務処理シーケンスの動作を示すフローチャートである。

【0109】業務処理シーケンス3は、業務監視中に、あるメッセージを受信した時（ステップ301）、そのメッセージコード3022が業務終了メッセージコードと同じであり（ステップ302）、待ち合わせ時間3023が0であり（＝待ち合わせを行わない）（ステップ303）、メッセージ判定情報3024によりそのメッセージが正常であることが判明したら（ステップ304）、定義したメッセージ数分の送信メッセージ詳細情報3026に基づいて送信する（ステップ305）。上述の業務開始メッセージ受信動作と同様に受信したメッセージが正常である場合には、送信メッセージ詳細情報の宛先情報3027の宛先に送信メッセージコード3028で、受信したインタフェースをインタフェース編集情報3029に基づいて編集する。編集されたインタフェースに基づいてメッセージを送信する（ステップ306）。最後に、業務番号3003に定義されている番号の業務監視を終了する（ステップ307）。

《メッセージ受信》さらに、ソフトウェア部品がメッセージを受信した場合の動作について説明する。図13は、本発明の第2の実施例のメッセージを受信した場合の業務処理シーケンスの動作を示すフローチャートである。ソフトウェア部品は、業務として監視されていなくても、メッセージの送受信は可能である。

【0110】業務処理シーケンス3は、あるメッセージを業務シーケンス3が受信したとき（ステップ401）、そのメッセージコードが受信メッセージ情報のメッセージコードと同じであり（ステップ402）、待ち合わせ時間3031が0であり（待ち合わせを行わない）（ステップ403）、メッセージ判定情報3032によりそのメッセージが正常であることが判明したら（ステップ404）業務処理シーケンス3は、送信メッセージ数3033で定義されているメッセージ数分のメッセージを送信メッセージ詳細情報3034に基づいて送信する（ステップ405）。上述の業務開始メッセージ受信動作と同様に受信したメッセージは正常である場

合に、宛先情報3035の宛先に送信メッセージコード3036で受信したインタフェースをインタフェース編集情報3037に基づいて編集する。さらに編集されたインタフェースに基づいてメッセージを送信する（ステップ406）。

【0111】上記のような動作を行うことにより、メッセージ送信先の変更、メッセージ内容の編集等を業務処理シーケンス3の定義体31を変更するだけでソフトウェア部品（プログラム）の変更を行う必要がない。また、同期制御、分配制御を業務処理シーケンスで行うことで、これらの変更も業務処理シーケンスの定義体を変更することにより可能である。また、メッセージの流れを業務シーケンス3で一つの業務として監視を行っているので、複数のメッセージの管理の変更も業務シーケンスの定義体の変更で行うことができる。

【0112】《メッセージマージ送信》次に、複数のソフトウェア部品からのそれぞれのメッセージをマージして送信する場合の例を説明する。

【0113】図14は、本発明の第2の実施例の他の例を示す。

【0114】同図に示す例は、ソフトウェア部品PKG-Bとソフトウェア部品PKG-Dからの2つのメッセージを待ち合わせして、1つのメッセージにマージしてソフトウェア部品PKG-Aに送信する場合を示す。また、送信するメッセージのインタフェースは、受信した2つのメッセージのインタフェースを編集したものである。

【0115】業務処理シーケンス3は、定義体31に待ち合わせるメッセージの数と待ち合わせ時間、待ち合わせた後に送信するメッセージの数とそれぞれの宛先と編集情報を定義する。この場合、業務シーケンス3がソフトウェア部品PKG-Bからのメッセージを受信してから、ソフトウェア部品PKG-Dからのメッセージを受信するまでに待ち合わせ時間を越えてしまった場合、業務処理シーケンス3は、ソフトウェア部品PKG-Aに対して異常時のメッセージを送信する。

【0116】図15は、本発明の第2の実施例の他の例を示すシーケンスチャートである。ソフトウェア部品PKG-Bは、まず、メッセージdを業務処理シーケンス3を宛先として送信する（ステップ41）。業務処理シーケンス3は、ソフトウェア部品PKG-Dからのメッセージが到来するまで送信処理を待機する（ステップ42）。ソフトウェア部品PKG-Dがメッセージeを業務処理シーケンス3に送信すると（ステップ43）、業務シーケンス3は、先にソフトウェア部品PKG-Bから到着してメッセージdと後からソフトウェア部品PKG-Dから到着したメッセージeをマージするメッセージ編集を行い（ステップ44）、編集されたメッセージfをソフトウェア部品PKG-Aに送信する（ステップ45）。さらに、業務処理シーケンス3は、ソフトウェ

ア部品PKG-B、PKG-Dから受信したインタフェース情報を編集して(ステップ46)、ソフトウェア部品PKG-Aに送信する(ステップ47)。

【0117】《メッセージ分散送信》次に、本実施例の他の例(その2)として、上記の方法の逆のパターンについて説明する。

【0118】図16は、本発明の第2の実施例の他の例(その2)を説明するための図である。同図に示す例は、ソフトウェア部品PKG-Bから送信された一つのメッセージを複写・編集して2つのメッセージとして、2つのソフトウェア部品PKG-A、PKG-Cの宛先に送信するものである。業務処理シーケンスの定義体にあるメッセージ受信に対して送信するメッセージの数とそれぞれの宛先と編集情報を定義することができる。

【0119】また、業務処理シーケンスの定義体に一連のメッセージの流れを一つの業務として定義することで、時間監視を行うことができる。業務開始メッセージを受信した時から業務終了メッセージの受信までを一つの業務とするが、業務開始メッセージを受信してから定義体31に定義した業務監視時間3004が過ぎるまでに業務終了メッセージを受信しなかった場合には、タイムアウトとして定義体31に定義してある異常時のメッセージを送信する。

【0120】図17は、本発明の第2の実施例の他の例(その2)のシーケンスチャートである。

【0121】ソフトウェア部品PKG-Bが業務処理シーケンス3にメッセージを送信する(ステップ51)。業務処理シーケンス3は、ソフトウェア部品PKG-Bからのメッセージを受信すると、メッセージを編集し(ステップ52)、編集されたメッセージを複写して(ステップ53)、ソフトウェア部品PKG-A及びソフトウェア部品PKG-Cに送信する(ステップ54、55)。次に、業務処理シーケンス3は、インタフェース情報を編集し(ステップ56)、メッセージと同様に、インタフェースをソフトウェア部品PKG-A及びPKG-Cに送信する(ステップ57、58)。

【0122】[第3の実施例]次に、本発明の第3の実施例として二重化システムの運転モード管理について説明する。本実施例では、ソフトウェア部品のアプリケーションソフトが自系の運転モードを意識することなく、アプリケーションソフトの業務切替を実現させるものである。運転モードに対応するアプリケーションソフト業務の有効/無効(ステータス)を定義しておき、このステータスにより運転モードが有効または無効の処理を行うものである。

【0123】図18は、本発明の第3の実施例を説明するための図である。同図(A)は、ステータスに定義してある運転モードを示すステータス定義情報であり、同図(B)は、業務シーケンス3が有する業務シーケンステーブルを示す。例えば、ステータス定義情報のステー

タス1では、業務シーケンステーブルの業務番号、が有効であると定義されている。このとき、有効となっている業務の処理aから処理b、業務の処理bから処理c、業務の処理cから処理dを実行する。即ち、ステータス1は、処理aから処理dを連続して主系モードで実行する。

【0124】図19は、本発明の第3の実施例のモード切替時の動作を説明するための図である。モードが主系から従系に切り替わった場合に、当該モードで実行していたステータスを無効とし、他のステータスを有効する。例えば、ステータス3が有効とされ、業務番号、の処理が主系で実行されていた場合に、モードが切り替わると、ステータス3は無効となる。従って、業務番号、の処理は実行されない。また、ステータス1を有効となるようにモードの切替が行われると、ステータス1の業務番号、の処理が有効となり実行される。この場合は、ステータス1の処理の流れ、aからdは主系で処理され、処理の中で運転モードをチェックする必要がない。

【0125】

【発明の効果】上述のように本発明のソフトウェア部品結合システムによれば、ソフトウェア構成や結合するソフトウェアの影響を受けずに、ソフトウェア部品自体で独立に開発や変更及び再利用が可能となる。また、分散/集中システムの構築及び構成の変更がソフトウェア部品内部のアプリケーションソフトを変更せずに、単に結合情報を変更するのみで可能となる。従って、結合相手のソフトウェアに変更が生じた場合でも結合相手のソフトウェアの変更に合わせて自ソフトウェアを変更する必要がないため、何度も再利用されるソフトウェアの管理が容易になる。例えば、売上システムにおいて、A営業所の売上プログラムとB営業所の売上プログラム、集計プログラムがあった場合に、A営業所の売上プログラムの内容について変更したい場合に、その変更に伴って集計プログラムの変更も必要となるケースにおいて、A営業所の売上プログラムの変更要素、及び集計プログラムの変更要素を結合情報に記入し、結合制御部で制御することにより、A営業所の売上プログラム及び集計プログラムは変更する必要がない。

【0126】また、本発明のメッセージ送受信システムによれば、メッセージ送受に必要な定義項目をプログラム内部に持たずに、業務処理シーケンスの定義体に定義することにより、いかなるメッセージ送受信条件の変更にもプログラムの修正の必要がないため、ソフトウェア部品の独立性を高めることができる。

【0127】さらに、本発明の運転モード管理システムによれば、運転モードに対応するソフトウェア部品の実行の有効・無効を業務シーケンステーブルに定義しておくことにより、ソフトウェア部品が運転モードを意識する必要がないので、単独CPUシステムから二重化CPU

システムへの移行や運転モードの追加等をソフトウェア部品の改造を行わなくとも可能となる。

【図面の簡単な説明】

【図 1】本発明の原理構成図（その 1）である。

【図 2】本発明の第 1 の実施例の概要を示す図である。

【図 3】本発明の第 1 の実施例のソフトウェア部品の構成図である。

【図 4】本発明の第 1 の実施例のイベント情報形式を示す図である。

【図 5】本発明の第 1 の実施例の結合情報の例を示す図である。

【図 6】本発明の第 1 の実施例の動作の概要を示すフローチャートである。

【図 7】本発明の第 1 の実施例の図 6 のフローチャートに対応する動作を説明するための図である。

【図 8】本発明の第 2 の実施例のメッセージ送受信システムの構成図である。

【図 9】本発明の第 2 の実施例のメッセージ送受信のシーケンスを示す図である。

【図 10】本発明の第 2 の実施例の業務シーケンスの定義体の構成図である。

【図 11】本発明の第 2 の実施例の業務開始メッセージを受信した場合の業務シーケンスの動作を示すフローチャートである。

【図 12】本発明の第 2 の実施例の業務終了メッセージを受信した場合の業務シーケンスの動作を示すフローチャートである。

【図 13】本発明の第 2 の実施例のメッセージを受信した場合の業務シーケンス動作を示すフローチャートである。

【図 14】本発明の第 2 の実施例の他の例（その 1）を説明するための図である。

【図 15】本発明の第 2 の実施例の他の例（その 1）のシーケンスチャートである。

【図 16】本発明の第 2 の実施例の他の例（その 2）を説明するための図である。

【図 17】本発明の第 2 の実施例の他の例（その 2）のシーケンスチャートである。

【図 18】本発明の第 3 の実施例を説明するための図である。

【図 19】本発明の第 3 の実施例のモード切替時の動作を説明するための図である。

【図 20】従来の第 1 の例を示す図である。

【図 21】従来の第 2 の例を示す図である。

【図 22】従来の第 2 の例において、プログラム修正により宛先を変更する例を示す図である。

【図 23】従来のシステムの運転モード管理を説明するための図である。

【符号の説明】

1 ソフトウェア部品

2 メッセージ

3 業務処理シーケンス

4 メッセージ

5 ソフトウェア部品

10 ソフトウェア部品

11、13 装置

14 メッセージ管理部

15 宛先対象表

20 制御手段

30 データ変換テーブル

31 業務シーケンスの定義体

40 共通参照情報

50 ステータス情報

60 業務シーケンステーブル

100 プロセス X のメッセージキュー

101～103、110～115 イベント

200 結合制御部

210 イベント結合ルーチン

300 リアルタイム情報・バッチ情報

400 結合情報

500 イベント情報

510 イベント種別

520 宛先情報

521 装置 ID

522 プロセス ID

523 ソフトウェア部品 ID

530 発信元情報

531 装置 ID

532 プロセス ID

533 ソフトウェア部品 ID

534 イベント番号

540 通知情報長

550 通知情報

600 応答イベント格納領域

1010 アプリケーションソフト

3001 業務数

3002 業務情報

3003 業務番号

3004 業務監視時間

3005 業務開始メッセージ情報

3006 待ち合わせ時間

3007 排他種別

3008 受信メッセージ

3009 受信メッセージ数

3010 受信メッセージ詳細情報

3011 受信メッセージコード

3012 メッセージ判定情報

3013 送信メッセージ情報

3014 送信メッセージ数

3015 送信メッセージ詳細情報

3016 宛先情報  
 3017 正常時情報  
 3018 異常時情報  
 3019 メッセージコード  
 3020 インタフェース編集情報  
 3021 メッセージコード  
 3022 受信メッセージコード  
 3023 待ち合わせ時間  
 3024 メッセージ判定情報  
 3025 送信メッセージ数  
 3026 送信メッセージ情報  
 3027 宛先情報

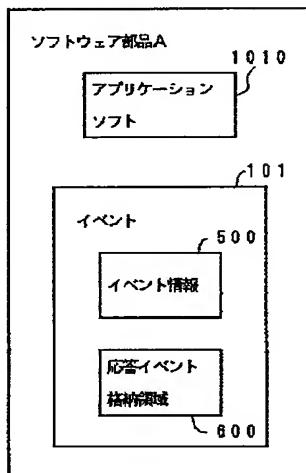
【図1】

3028 正常時メッセージコード  
 3029 正常時インタフェース情報  
 3030 受信メッセージ数  
 3031 待ち合わせ時間  
 3032 メッセージ判定情報  
 3033 送信メッセージ数  
 3034 受信メッセージ詳細情報  
 3035 宛先情報  
 3036 正常時メッセージコード  
 3037 正常時インタフェース編集情報  
 3040 業務終了メッセージ情報  
 3041 メッセージ情報

【図2】

【図3】

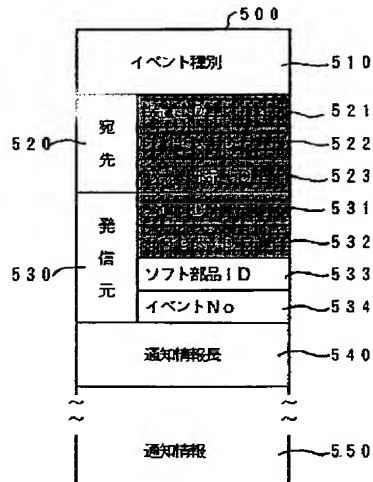
本発明の第1の実施例のソフトウェア部品の構成図





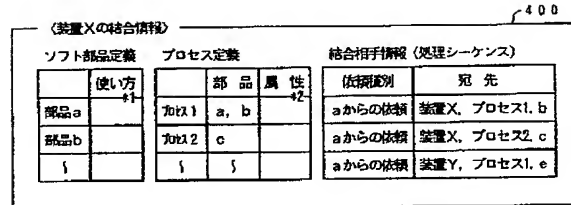
【図4】

本発明の第1の実施例のイベント情報形式を示す図



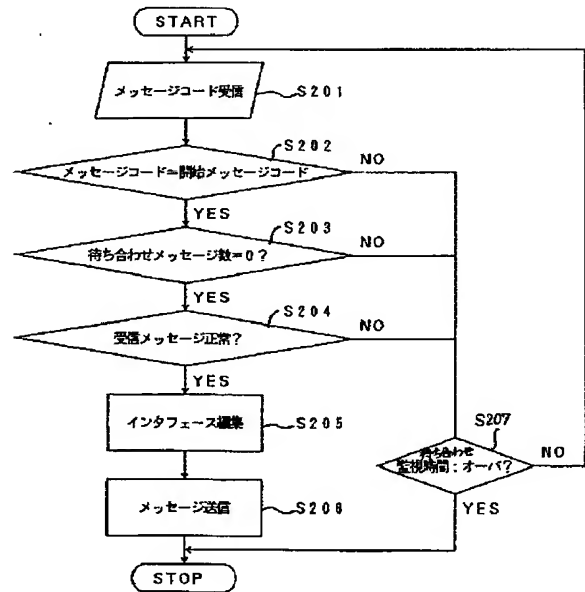
【図5】

本発明の第1の実施例の結合情報の例を示す図



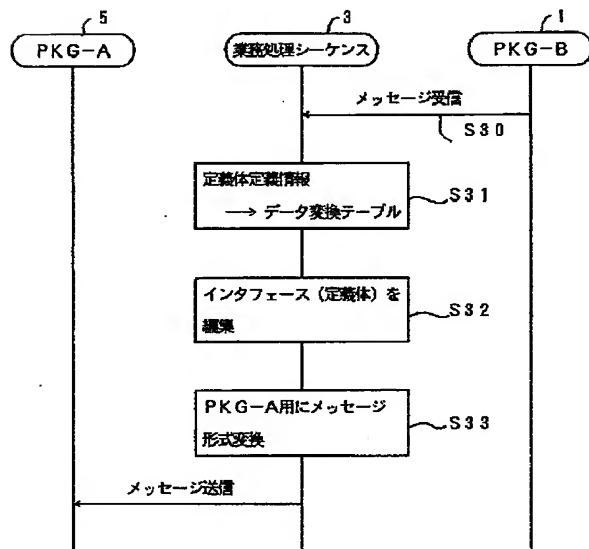
【図11】

本発明の第2の実施例の業務開始メッセージを受信した場合の業務シーケンスの動作を示すフローチャート



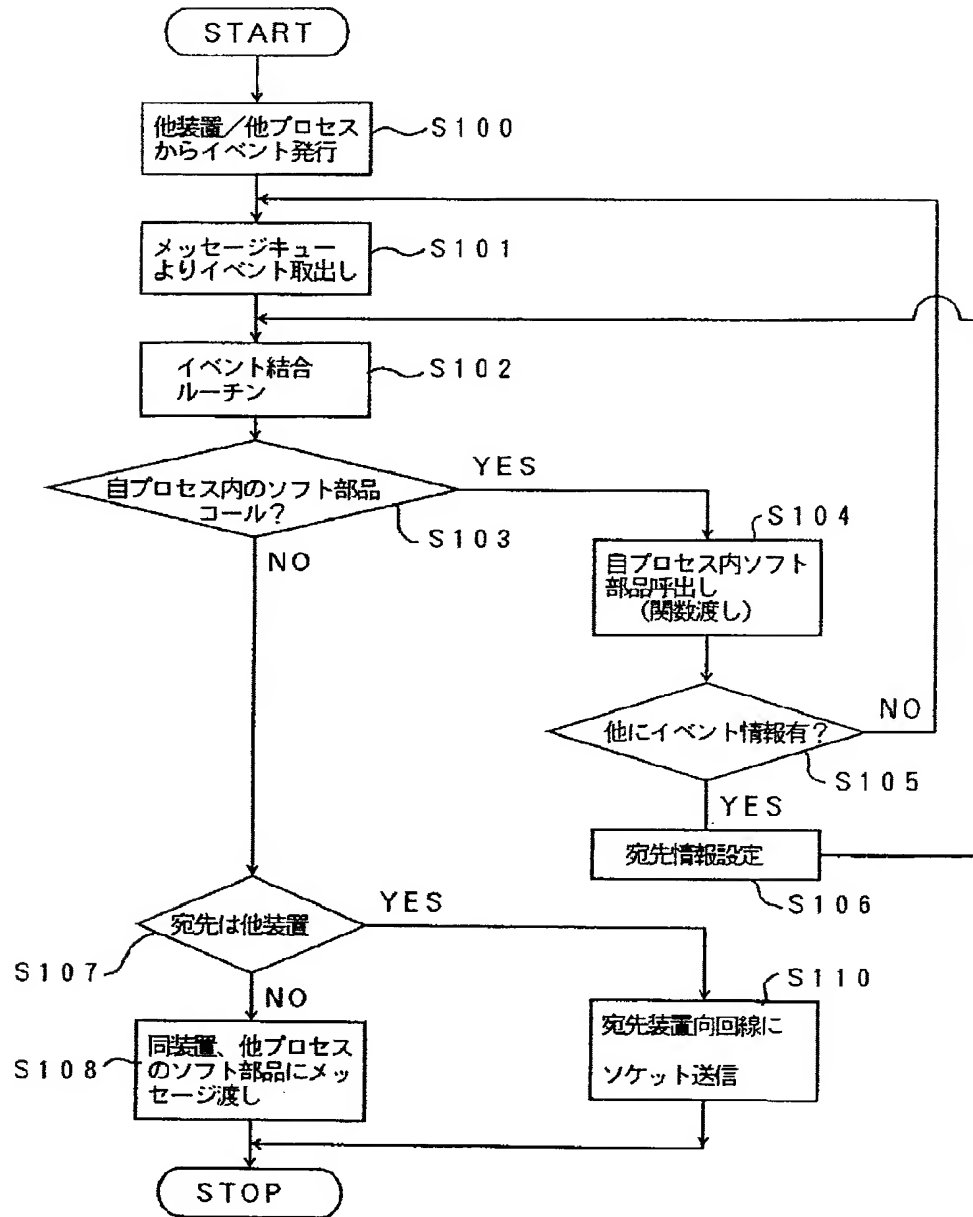
【図9】

本発明の第2の実施例のメッセージ送受信のシーケンスを示す図

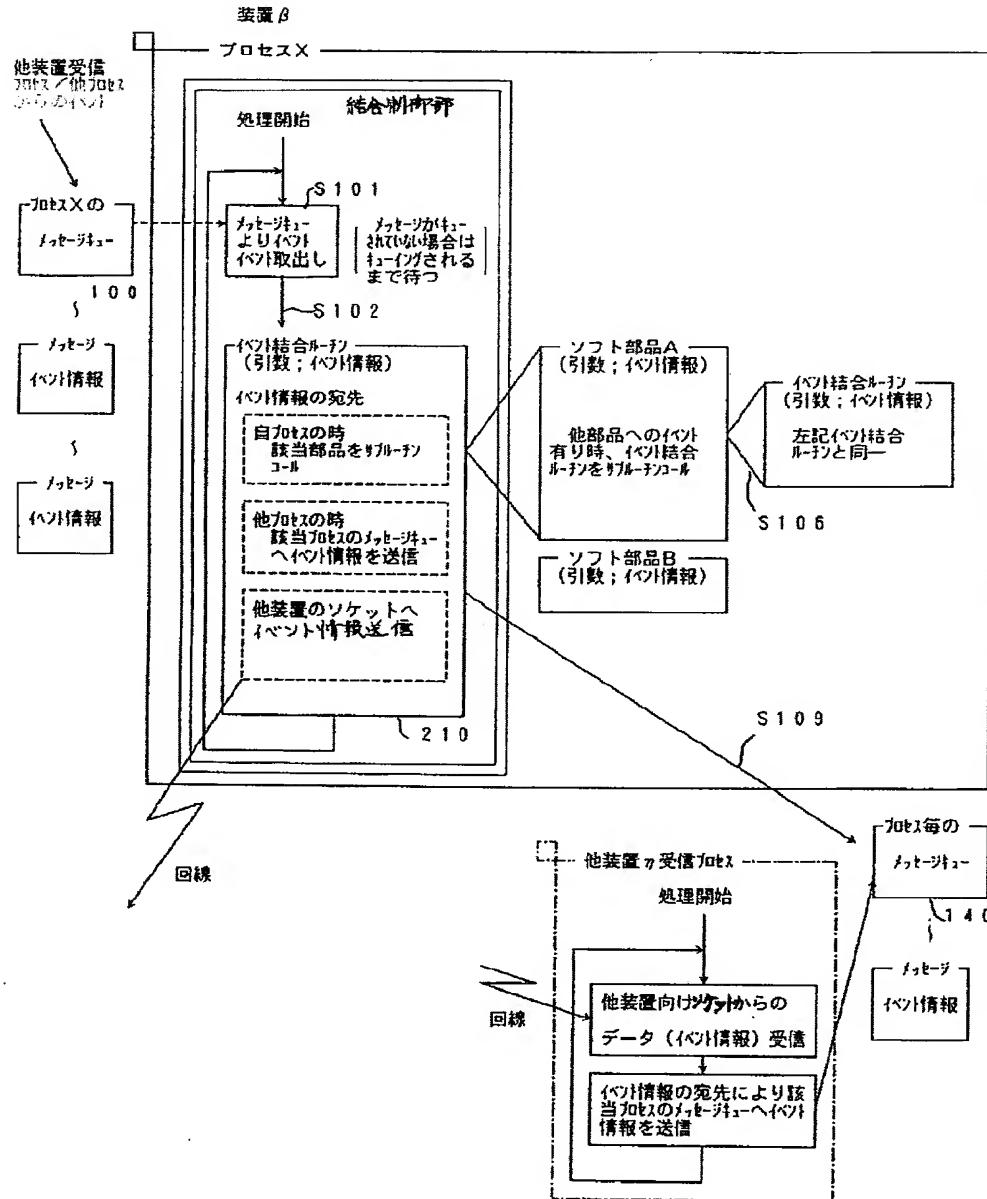


【図6】

本発明の第1の実施例の動作の概要を示すフローチャート

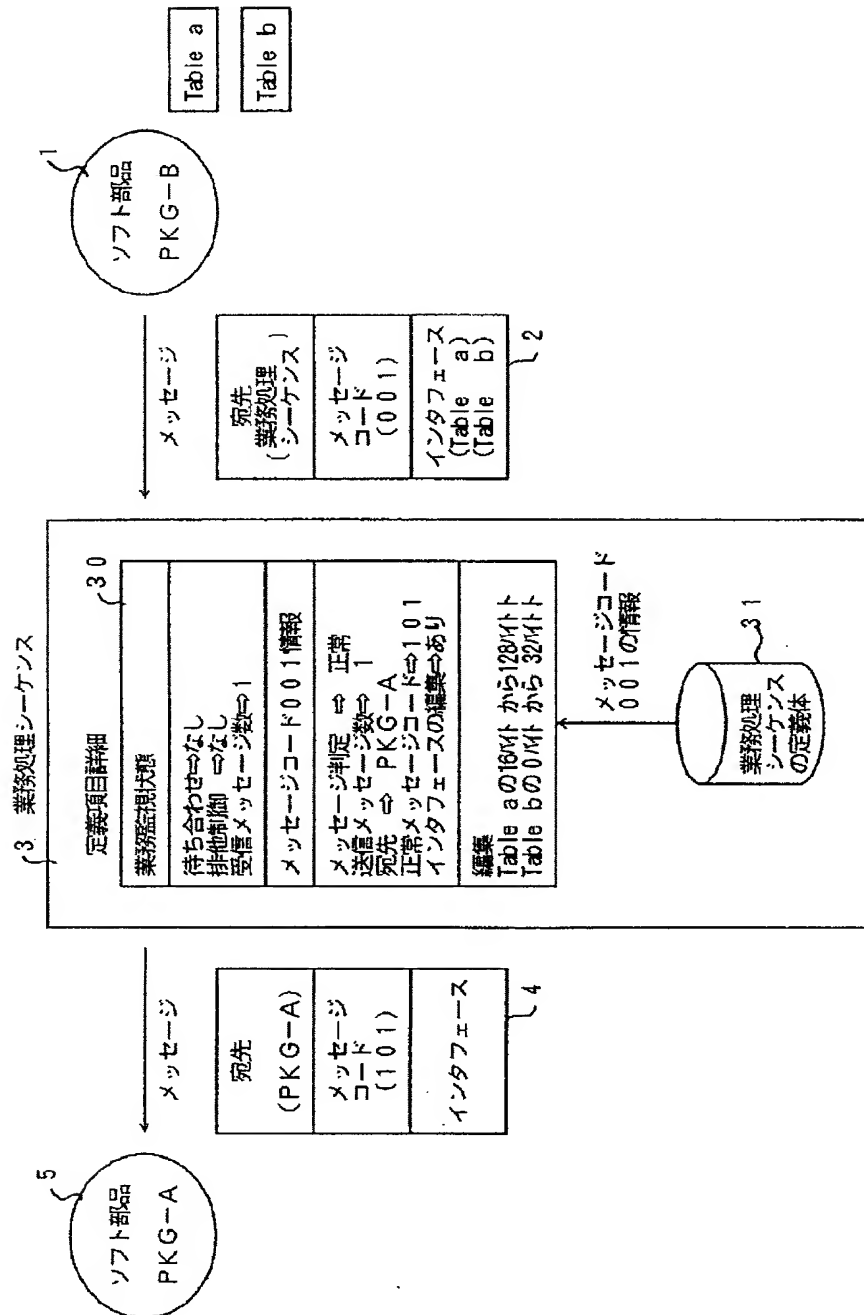


本発明の第１の実施例の図６のフローチャートに対応する動作を説明するための図



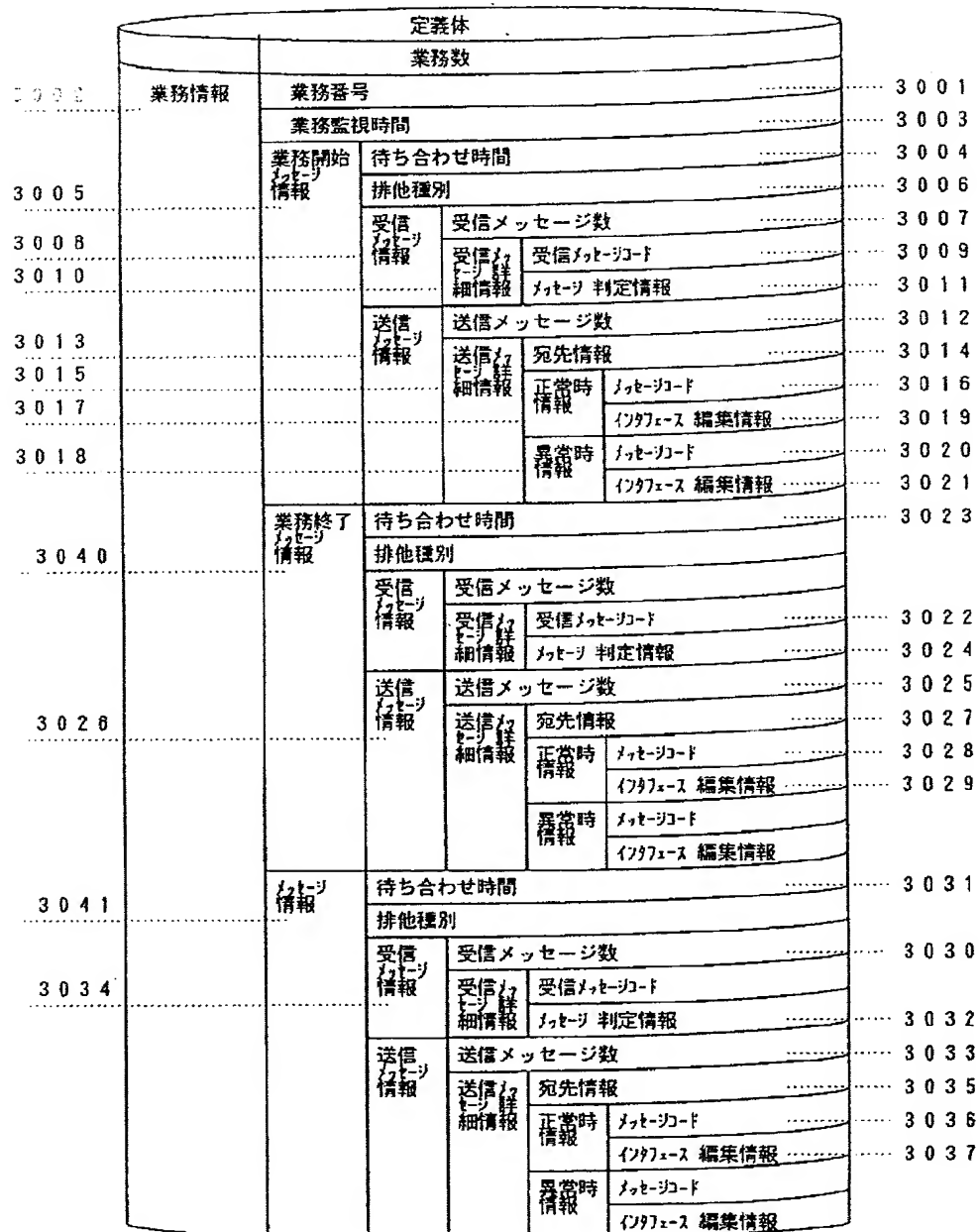
【図8】

本発明の第2の実施例のメッセージ送受信システムの構成図



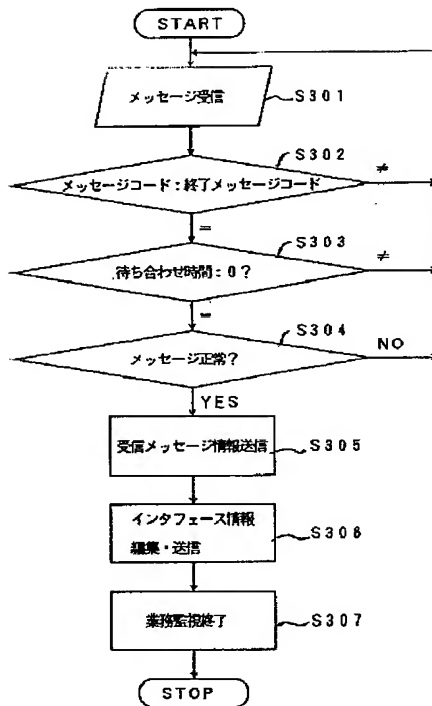
【図10】

本発明の第2の実施例の業務シーケンスの定義体の構成図



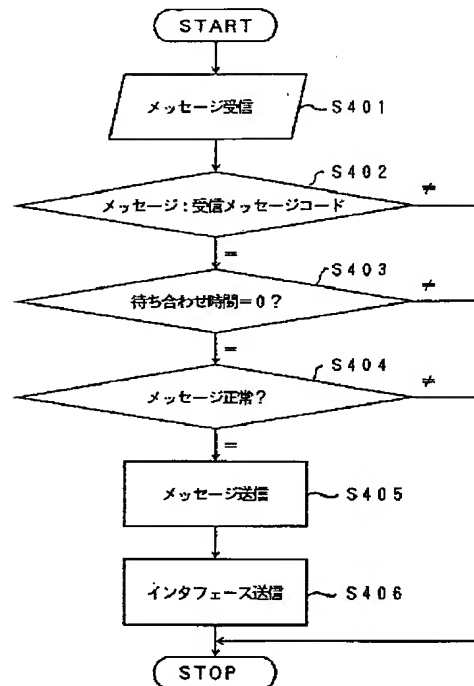
【図12】

本発明の第2の実施例の業務終了メッセージを受信した場合の  
業務処理シーケンスの動作を示すフローチャート



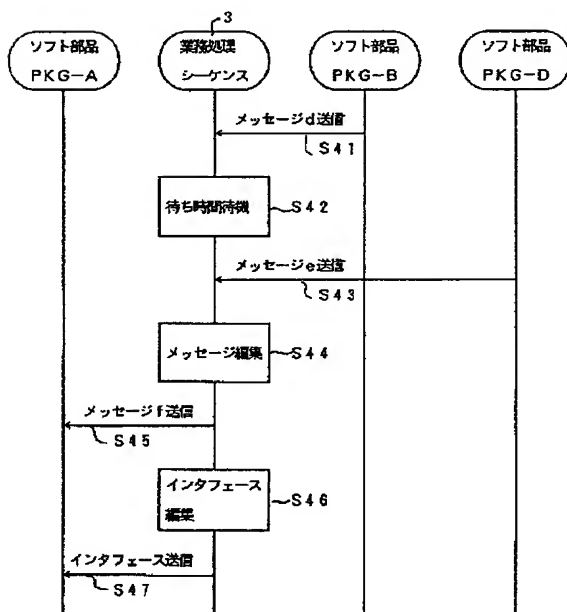
【図13】

本発明の第2の実施例のメッセージを受信した場合の  
業務シーケンスの動作を示すフローチャート



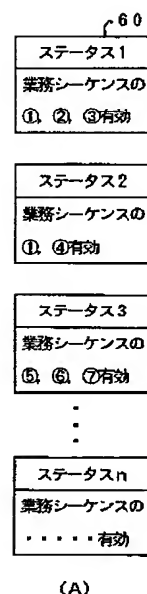
【図15】

本発明の第2の実施例の他の例を示すシーケンスチャート



【図18】

本発明の第3の実施例を説明するための図



(A)

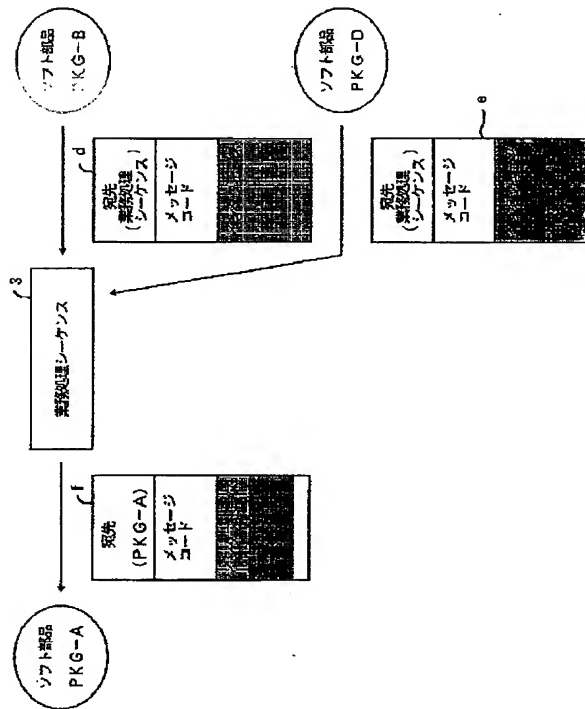
業務シーケンステーブル 50

| 業務No | 業務シーケンス   | 有効/無効 |
|------|-----------|-------|
| ①    | 処理a → 処理b | ・     |
| ②    | 処理b → 処理c | ・     |
| ③    | 処理c → 処理d | ・     |
| ④    | 処理b → 処理f | ・     |
| ⑤    | 処理g → 処理f | ・     |
| ⑥    | 処理h → 処理a | ・     |
| ⑦    | 処理a → 処理i | ・     |
| ・    | ・         | ・     |
| ・    | ・         | ・     |
| ・    | ・         | ・     |

(B)

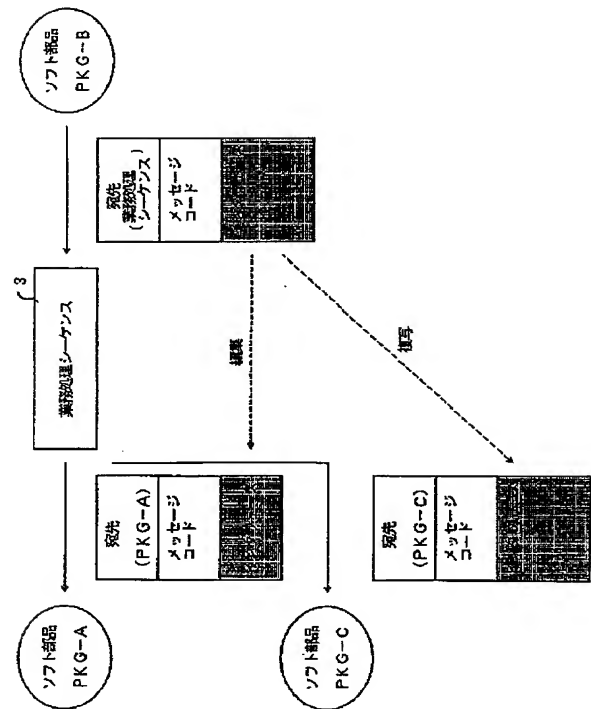
【図14】

本発明の第2の実施例の他の例（その1）を説明するための図



【図16】

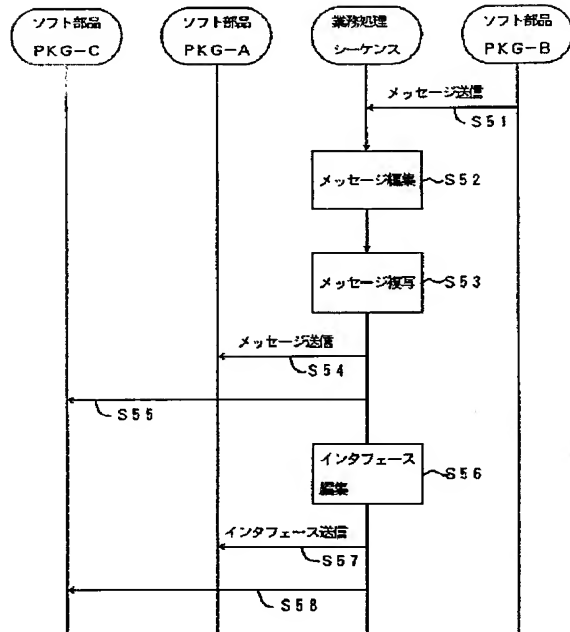
本発明の第2の実施例の他の例（その2）を説明するための図





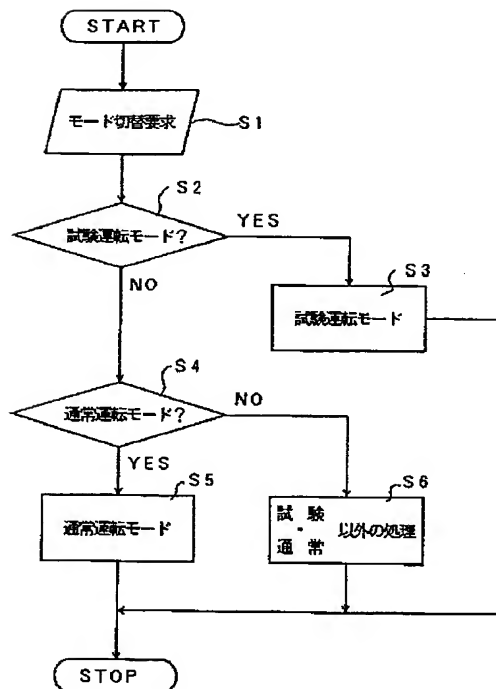
【図17】

本発明の第2の実施例の他の例を（その2）のシーケンスチャート



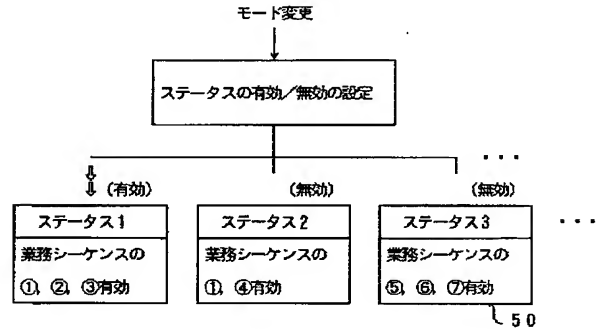
【図23】

従来システムの運転モード管理を説明するための図



【図19】

本発明の第3の実施例のモード切替時の動作を説明するための図

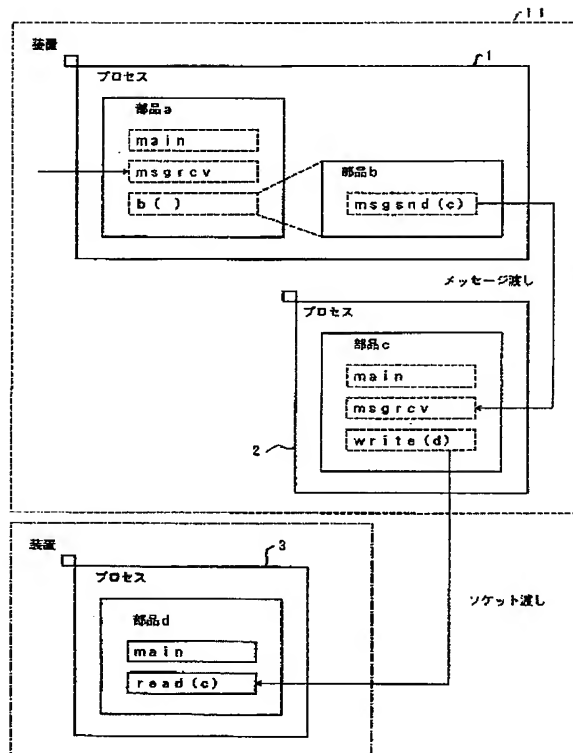


業務シーケンステーブル

| 業務No. | 業務シーケンス   | 有効/無効 |
|-------|-----------|-------|
| ①     | 処理a → 処理b | 有効    |
| ②     | 処理b → 処理c | 有効    |
| ③     | 処理c → 処理d | 有効    |
| ④     | 処理b → 処理f | 無効    |
| ⑤     | 処理g → 処理f | 無効    |
| ⑥     | 処理h → 処理a | 無効    |
| ⑦     | 処理a → 処理i | 無効    |
| ⋮     | ⋮         | ⋮     |
| ⋮     | ⋮         | ⋮     |
| ⋮     | ⋮         | ⋮     |

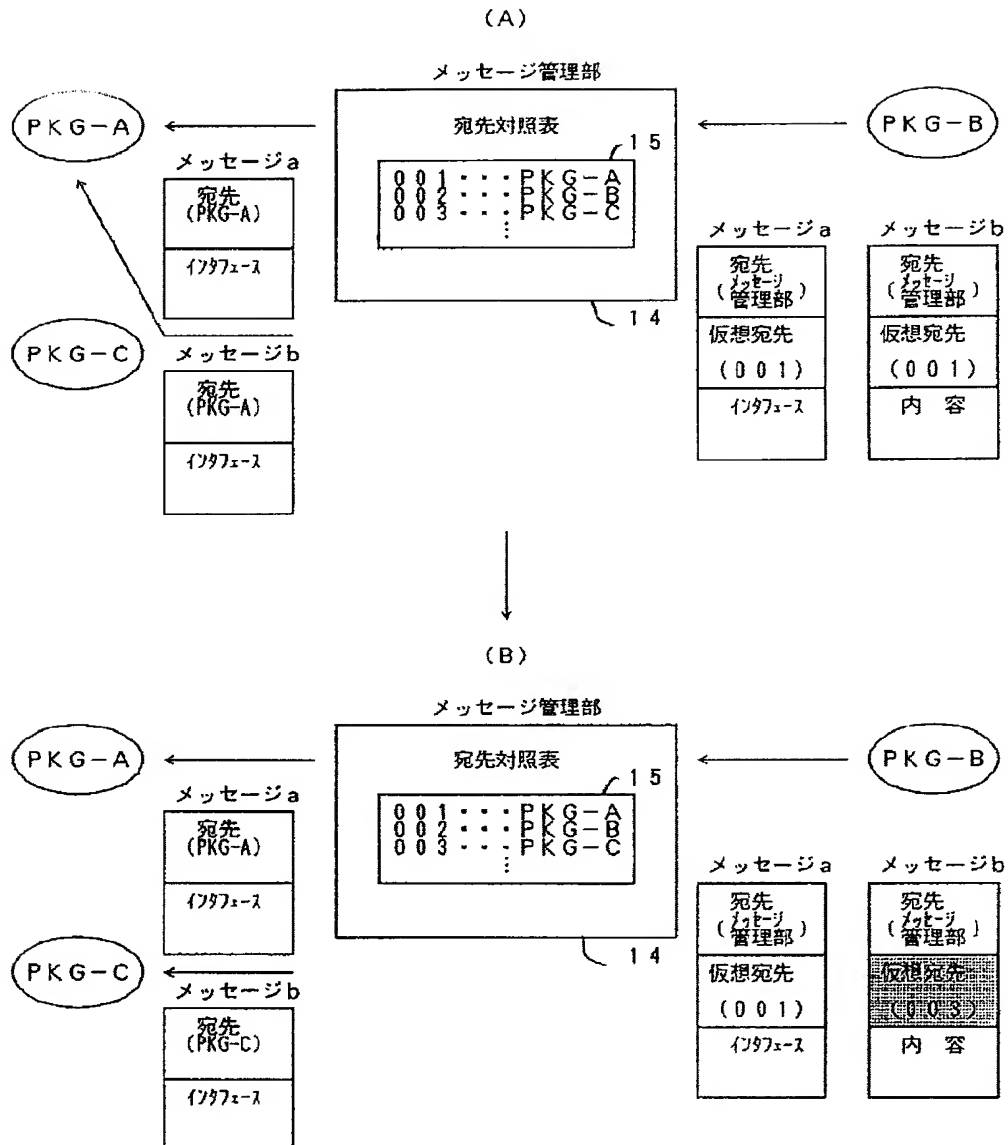
【図20】

従来の第1の例を示す図



【図21】

従来の第2の例を示す図



【図22】

従来の第2の例においてプログラム修正により宛先を変更する例を示す図

